

**CLAIMS:**

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

- 1                   1. A method for performing a background code update of a current code  
2 image with an incoming code image in an embedded system, the method comprising the  
3 steps of:
  - 4                   (a) executing the current code image in the embedded system;
  - 5                   (b) executing one or more code update routines from the incoming code  
6 image to update the current code image with the incoming code image; and
  - 7                   (c) executing a task switching function from the current code image to  
8 switch microprocessor control from executing the one or more code update routines of  
9 the incoming image to execute a function in the current code image.
- 10  
1                   2. The method according to Claim 1, wherein the method further  
2 comprises a step of retrieving an offset from the incoming code image for the one or  
3 more code update routines in the incoming code image.
- 1                   3. The method according to Claim 1, wherein the method further  
2 comprises a step of retrieving an offset from the current code image of a task switching  
3 function.
- 1                   4. The method according to Claim 1, wherein the method further  
2 comprises a step of loading all or part of the incoming code image into random access  
3 memory for execution.
- 1                   5. The method according to Claim 1, wherein the method further  
2 comprises receiving the incoming code image into the embedded system via an  
3 input/output interface.

1                   6. The method according to Claim 1, wherein the method further  
2 comprises the steps of:  
3                   providing a plurality of programmable memory devices for storing  
4 copies of the current code image;  
5                   executing a copy of the current code image from one programmable  
6 memory device; and  
7                   updating a copy of the current code image in other programmable  
8 memory device with the incoming code image.

1                   7. The method according to Claim 3, wherein the method further  
2 comprises a step of testing the offset of the task switching function for validity before  
3 executing the task switching function.

1                   8. The method according to Claim 1, wherein the method further  
2 comprises the steps of:  
3                   yielding microprocessor control by the executing function upon a task  
4 switching event; and  
5                   switching microprocessor control to continue executing the one or more  
6 code update routines to update the current code image with the incoming code image.

1                   9. The method according to claim 8, wherein the method further  
2 comprises a step of continuing to switch microprocessor control between the one or  
3 more code update routines of the incoming code image and one or more functions of the  
4 current code image until the background code update completes.

1                   10. The method according to Claim 1, wherein the task switching event  
2 is one selected from the group consisting of: round robin task switching; event driven  
3 task switching; and time slice task switching.

1                   11. The method according to Claim 8, wherein the task switching event  
2 is one selected from the group consisting of: round robin task switching; event driven  
3 task switching; and time slice task switching.

1                   12. The method according to Claim 1, wherein the method further  
2 comprises a step of resetting the embedded system upon completion of the background  
3 code update.

1                   13. An embedded system for performing a background code update of a  
2 current code image with an incoming code image, the system comprising:

3                   a first programmable memory device for storing the current code image;

4                   a microprocessor for executing the current code image in the embedded  
5 system and for executing one or more code update routines to update the current code  
6 image with the incoming code image; and

7                   a task switching means for executing a task switching function in the  
8 current image to switch microprocessor control from executing the one or more code  
9 update routines of the incoming image to execute a function in the current code image.

10

1                   14. The embedded system according to Claim 13, wherein the embedded  
2 system further comprises a random access memory for loading all or part of the  
3 incoming code image for execution by the microprocessor.

1                   15. The embedded system according to Claim 14, wherein the embedded  
2 system further comprises an input/output interface for receiving the incoming code  
3 image into the embedded system.

1                   16. The embedded system according to Claim 13, wherein the embedded  
2 system further comprises a second programmable memory device for storing a copy the  
3 current code image, wherein the microprocessor executes the current code image from  
4 the first programmable memory device and updates the copy of the current code image  
5 in the second programmable memory device with the incoming code image.

1                   17. The embedded system according to Claim 13, wherein the incoming  
2 code image instructs the microprocessor to further test the offset of the task switching  
3 function for validity before executing the task switching function.

1                   18. The embedded system according to Claim 13, wherein the task  
2 switching means further switches microprocessor control from the executing function to  
3 continue executing the one or more code update routines to update the current code  
4 image with the incoming code image.

1                   19. The embedded system according to claim 18, wherein the switching  
2 function instructs the microprocessor to switch control between the one or more code  
3 update routines of the incoming code image and one or more functions of the current  
4 code image until the background code update completes.

1                   20. The embedded system according to Claim 13, wherein the task  
2 switching means is one selected from the group consisting of: round robin task  
3 switching; event driven task switching; and time slice task switching.

1                   21. The embedded system according to Claim 18, wherein the task  
2 switching means is one selected from the group consisting of: round robin task  
3 switching; event driven task switching; and time slice task switching.

1                   22. The embedded system according to Claim 13, wherein the embedded  
2 system further comprises a bootloader for instructing the microprocessor to execute the  
3 current code image and the one or more code update routines of the incoming code  
4 image, and to reset the embedded system upon completion of the background code  
5 update.

1                   23. The embedded system according to Claim 15, wherein the embedded  
2 system comprises a bus for interconnecting one or more system components including  
TUC920010094US1

3 the microprocessor, the random access memory, the first programmable memory  
4 device, and the input/output interface.

1 24. The embedded system according to Claim 23, wherein one or more  
2 of the system components form a part of an integrated microprocessor.

1 25. The embedded system according to Claim 22, wherein the  
2 programmable memory device comprises a boot sector for storing the bootloader.

1 26. The embedded system according to Claim 22, wherein the bootloader  
2 tests the integrity of the current code image before instructing the microprocessor to  
3 execute it.

1 27. The embedded system according to Claim 22, wherein the bootloader  
2 is enabled to check for availability of a code update and if the code update is available  
3 to initiate the code update.

1 28. The embedded system according to Claim 13, wherein the current  
2 code image and the incoming code image include offsets within each respective image  
3 for code update routines and a task switching function.

1 29. The embedded system according to Claim 28, wherein the each  
2 respective code image comprises a header area for storing the offsets for the code  
3 update routines and the task switching function.

1 30. The embedded system according to Claim 13, wherein the offsets for  
2 the code update routines and the task switching function are stored at predetermined  
3 locations within each respective code image.

1 31. A storage automation library comprising an embedded system, the  
2 embedded system comprising:

TUC920010094US1

3 a first programmable memory device for storing the current code image;  
4 a microprocessor for executing the current code image in the embedded  
5 system and for executing one or more code update routines to update the current code  
6 image with the incoming code image; and  
7 a task switching means for executing a task switching function in the  
8 current image to switch microprocessor control from executing the one or more code  
9 update routines of the incoming image to execute a function in the current code image.

1 32. A program storage device, tangibly embodying a program of  
2 instructions executable by a machine to perform a method for performing a background  
3 code update of a current code image with an incoming code image in an embedded  
4 system, the method comprising the steps of:

5 (a) executing the current code image in the embedded system;  
6 (b) executing one or more code update routines from the incoming code  
7 image to update the current code image with the incoming code image; and  
8 (c) executing a task switching function from the current code image to  
9 switch microprocessor control from executing the one or more code update routines of  
10 the incoming image to execute a function in the current code image.

1 33. The program storage device according to Claim 32, wherein the  
2 method further comprises a step of retrieving an offset from the incoming code image  
3 for the one or more code update routines in the incoming code image.

1 34. The program storage device according to Claim 32, wherein the  
2 method further comprises a step of retrieving an offset from the current code image of a  
3 task switching function.

1 35. The program storage device according to Claim 32, wherein the  
2 method further comprises a step of loading all or part of the incoming code image into  
3 random access memory for execution.

1                   36. The program storage device according to Claim 32, wherein the  
2 method further comprises receiving the incoming code image into the embedded system  
3 via an input/output interface.

1                   37. The program storage device according to Claim 32, wherein the  
2 method further comprises the steps of:  
3                   providing a plurality of programmable memory devices for storing  
4 copies of the current code image;  
5                   executing a copy of the current code image from one programmable  
6 memory device; and  
7                   updating a copy of the current code image in the other programmable  
8 memory device with the incoming code image.

1                   38. The program storage device according to Claim 34, wherein the  
2 method further comprises a step of testing the offset of the task switching function for  
3 validity before executing the task switching function.

1                   39. The program storage device according to Claim 32, wherein the  
2 method further comprises the steps of:  
3                   yielding microprocessor control by the executing function upon a task  
4 switching event; and  
5                   switching microprocessor control to continue executing the one or more  
6 code update routines to update the current code image with the incoming code image.

1                   40. The program storage device according to claim 39, wherein the  
2 method further comprises a step of switching microprocessor control between the one or  
3 more code update routines of the incoming code image and one or more functions of the  
4 current code image until the background code update completes.

1                   41. The program storage device according to Claim 34, wherein the task  
2 switching event is one selected from the group consisting of: round robin task  
3 switching; event driven task switching; and time slice task switching.

1                   42. The method according to Claim 39, wherein the task switching event  
2 is one selected from the group consisting of: round robin task switching; event driven  
3 task switching; and time slice task switching.

1                   43. The program storage device according to Claim 32, wherein the  
2 method further comprises a step of resetting the embedded system upon completion of  
3 the background code update.

1                   44. A method for performing a background code update of a current code  
2 image with an incoming code image in an embedded system, the method comprising the  
3 steps of:

- 4                   (a) executing the current code image in the embedded system;  
5                   (b) retrieving an offset from the incoming code image of one or more  
6 code update routines in the incoming code image;  
7                   (c) executing the one or more code update routines to update the current  
8 code image with the incoming code image;  
9                   (d) retrieving an offset from the current code image of a task switching  
10 function upon a task switching event; and  
11                   (e) executing the task switching function to switch microprocessor  
12 control from executing the one or more code update routines of the incoming image to  
13 execute a function in the current code image.